**EJERCICIOS PHP**

1. Realizar el ejercicio dos con un formulario y otor fichero php que nos visualiza la letra del DNI

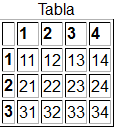
a) Version 1 con Metodo GET

b) Version 2 con Metodo POST

c) Version 3 con href

2.Realizar un unico programa que nos presente un formulario de petición de DNI cuando el DNI no exista y que nos visualizar la letra del DNI cuando el DNI si exista.

3.Realizar un fichero php que genere la siguiente tabla dinámicamente



4. Realizar una pagina autoprocesada en la que se presentan las tablas de multiplicar. Si no le pasamos numero de filas nos presenta las 10 tablas pero si nos pasan como parámetro el número de filas

Probamos la directiva request\_order ‘GP’ y la superglobal $\_REQUEST[‘valor’]

Hacemos un formulario al que le pasamos un dni con action='letra\_dni.php**?numero=50453277**' y method=’POST’

Observamos que aunque pasamos el parametro con GET solo coge el del POST

Metemos en .htaccess la directiva

php\_value request\_order 'PG'

Hacemos el mismo ejercicio y recibe con mayor prioridad lo del get

6.

**EJERCICIO: LCG**

En este ejercicio se propone al alumno desarrollar un generador de números aleatorios basado en el algoritmo *Linear Congruential Generator* (LCG). Este sencillo algoritmo utiliza la siguiente expresión:

Xn+1=(aXn+c) mod m

Donde:

* Xn es la secuencia de números aleatorios
* X0 es la semilla que deber un entero comprendido en [0,m)
* m se denomina módulo y debe ser un entero positivo
* a se denomina multiplicador y debe ser un entero comprendido en (0,m)
* c se denomina incremento y deber ser un entero comprendido en [0,m)

Para no provocar un desbordamiento de enteros, se propone utilizar los siguientes parámetros:

* m=32749
* a=219
* c=0

7. Aprovechando la función anterior generar una matriz de 20\*20 con 400 números aleatorios que no estén repetidos

8.

Obtener una función que baraja una baraja española

Version:

Global y por referencia

9.

Escribir un programa que muestre un mensaje diferente en función de la hora del día. Por ejemplo si es por la mañana muestra “Buenos Días”

10.

Crear un tablero de ajedrez mediante una tabla utilizando un bucle for anidado dentro de otro.

11.

Dado un número entero almacenado en la variable $numero, escribir el código necesario para asignar a la variable $es\_primo el valor TRUE en caso de que ese número sea primo, y el valor FALSE en caso contrario.

Se sugiere ir probando a dividir ese número entre todos los inferiores a la mitad de él (si no es divisible de forma exacta por ningún número inferior a su mitad, ya sólo puede serlo por sí mismo) empezando por 2 y continuando sólo con los impares (si no es divisible entre 2 no lo será tampoco entre ningún otro par), y en cuanto el módulo de alguna de esas divisiones sea nulo, determinar que no es primo y abandonar el bucle con un break.

Si se alcanza el final del bucle, es número debe ser primo.

12.

Para garantizar la cadena de frío de cierto número de cajas con vacunas en un transporte se ha instalado en cada una de ellas un sensor que ha ido recogiendo la evolución de la temperatura.

Los datos recogidos por los sensores están disponibles en el array $temperaturas del código que se muestra a continuación.

|  |  |
| --- | --- |
| 001 | <?php |
| 002 | $temperaturas=array(); |
| 003 | $temperaturas['Caja\_1']=array(1,1,2,3,2,1,2,3,3,3,2,1,3,4); |
| 004 | $temperaturas['Caja\_2']=array(0,0,3,2,4,3,2,0,1,2,3,4,2,1); |
| 005 | $temperaturas['Caja\_3']=array(3,1,2,3,5,2,2,0,1,2,3,4,2,1); |
| 006 | $temperaturas['Caja\_4']=array(2,2,2,3,5,2,3,2,0,1,2,3,0,1); |
| 007 | $temperaturas['Caja\_5']=array(0,3,2,3,5,2,3,2,0,1,2,3,0,1); |
| 008 | ?> |

Añada el código necesario para que se indique qué cajas han estado en algún momento sometidas a una temperatura superior a 4º.

13.

Crear una función que nos cree una tabla. El prototipo de la función debe ser como el que aparece a continuación

crear\_tabla(4, 6,'width: 60px;','height: 40px;','background: pink;','border: 3px dashed blue;');

Los parámetros en rojo son obligatorios y los demás no lo son, por tanto, cuando no los pasan tomara el alto y ancho de 70 el color azul y el borde negro.

14.

Escribir una función personalizada llamada buscar($aguja,$pajar) que devuelva un array con la posición de todas las ocurrencias de aguja en pajar, o el valor FALSE en caso de que no haya ninguna.

Probarla con la llamada buscar ('Ana', 'Ana Irene Palma').

15.

Hacer una función que compruebe la entrada de datos siguiendo las siguientes especificaciones

**FUNCION DE ENTRADA DE DATOS**

* La primera situación que los programas deben tener en cuenta es que los controles no contengan ningún valor.
* Pero si el usuario no escribe nada en el formulario, aunque el programa funcione correctamente, la respuesta del programa puede confundir al usuario
* (isset) Un problema más grave es que el programa suponga que existe un control que en realidad no le ha llegado.
* Eso puede ocurrir, por ejemplo, en el caso de las casillas de verificación y los botones radio, ya que los formularios envían el control solamente cuando se han marcado.
* **Seguridad en las entradas**

Un usuario puede insertar código html en la entrada de un formulario, lo que puede acarrear comportamientos inesperados y riesgos de seguridad. El siguiente ejemplo solamente perjudicaría al aspecto de la página

<strong>prueba</strong>

Probar un <body onload="alert('Hola')">

### Eliminar etiquetas: strip\_tags($cadena)

Una manera de resolver el problema anterior es utilizar la función [strip\_tags($cadena)](http://www.php.net/manual/es/function.strip-tags.php), que devuelve la cadena sin etiquetas, como muestra el siguiente ejemplo.

Aunque hay que tener en cuenta que esta función elimina cualquier cosa que se interprete como una etiqueta, es decir, que empiece por "<"

### Eliminar espacios en blanco iniciales y finales: trim($cadena)

RESUMEN DE LA FUNCION

La función tendrá como argumento el nombre del control que se quiere recibir y devolverá el valor recibido (o una cadena vacía si el control no se ha recibido).

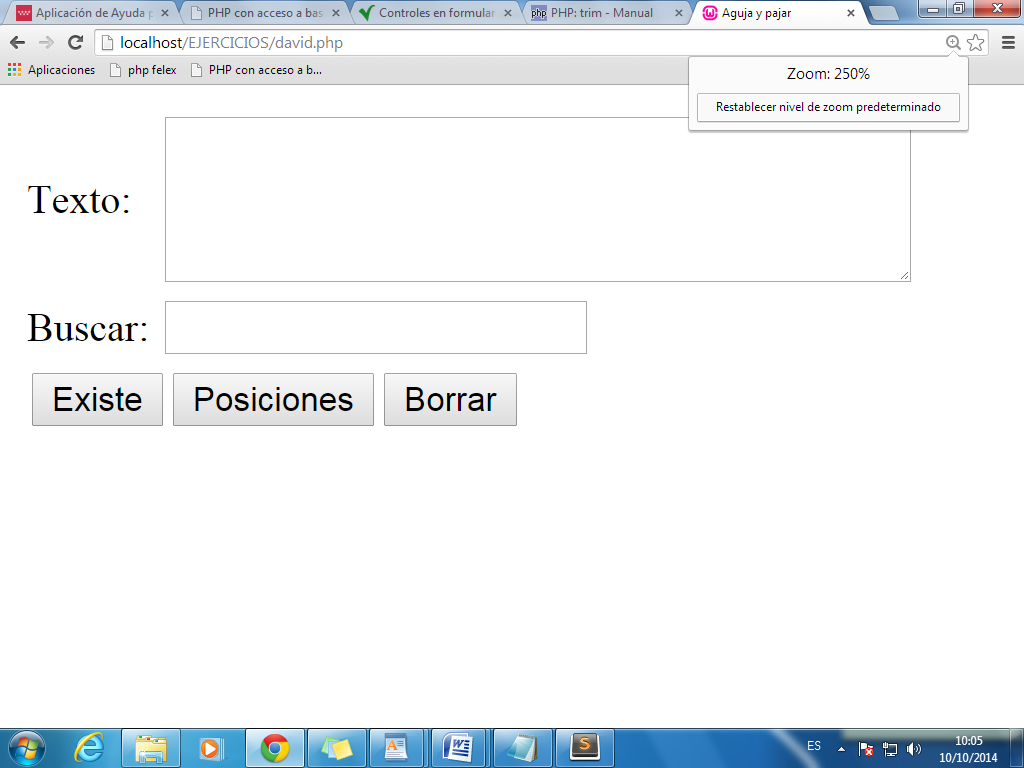
Para tratar los datos recibidos se aplicarán únicamente las funciones htmlspecialchars() y trim()

De esta manera, si se reciben etiquetas (<>), las etiquetas no se borrarán, sino que se conservarán como texto.  
Si se quisieran borrar las etiquetas, habría que aplicar las funciones strip\_tags(), trim(), y htmlspecialchars()

* Una vez definida la función, al comienzo del programa se almacenarán en variables los datos devueltos por la función .
* En el resto del programa se trabaja con las variables.

16.

Version 2 del ejercicio 14



17.

Vamos a realizar un formulario que nos permita seleccionar si hacer la versión del ejercicio 14 , ejercicio 16, el 1, 3 o el 5

Usar selección multiple e include

18.



Realizar un programa php que nos visualice todos los datos introducidos

**19.**

En este ejercicio vamos a crear dos funciones de validación del lado del servidor que pueden serle de utilidad frecuentemente (recuerde que para validar las fechas ya disponemos de la función checkdate).

1. Cree una función llamada is\_nif(*$cadena*) que compruebe si cadena es:
   1. Un número seguido de una única letra.
   2. La letra es la correspondiente a ese número.
2. Cree una función llamada is\_email(*$cadena*) que compruebe si cadena:
   1. Contiene un único signo @.
   2. Contiene un único punto por detrás de ese signo @.
   3. Contiene caracteres antes de la @, entre la @ y el punto, y detrás del punto.

20.

Cree en calendario\_funciones.php una función llamada **formulario\_evento()** que reciba como argumentos opcionales el año, el mes, el día, la hora, el minuto, un campo id, el título, y el comentario (por este orden).

Estos datos provienen del formulario presentado posteriormente.

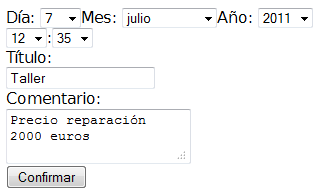
Cuando un parámetro no es pasado desde el formulario, excepto el id, incluya en el formulario un campo solicitándolo (si no se han omitido, deberán mostrarse en los campos del formulario).

La función deberá comprobar que los argumentos recibidos constituyen una fecha válida (ckeckdate si la fecha está completa, y si está incompleta el año deberá estar comprendido entre 2000 y 2037, y el mes entre 1 y 12), que la hora es válida (hora entre 0 y 23, y minuto entre 0 y 55 a múltiplos de 5), y que los otros dos argumentos son de tipo string y el id de tipo numérico (no NULL).

El formulario deberá contener además un campo oculto cuyo valor sea el id, y un botón submit para confirmar.

El action será procesar.php?operacion=confirmar y el método POST

El archivo procesar.php lo crearemos utilizando la función **formulario\_evento()**



21. Crear micebecera.php que pretende visualizar la cabecera con el siguiente aspecto

La clase Mi\_Cabecera, tendrá su constructor y el método dibujar. Esta clase debe ser incluida en micanecera.php que será el programa que utiliza la clase creada



22. Crear un programa que nos visualiza en una tabla el arrray que contiene los meses del año.

Utilizar para ello array\_walk($meses,’escribir\_tabla’);

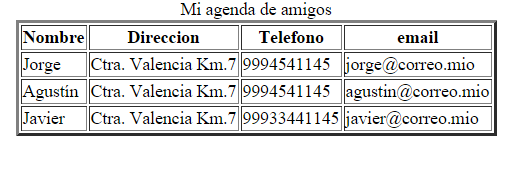
Generar la siguiente salida en función del mes que sea sacaremos sus días.



23.

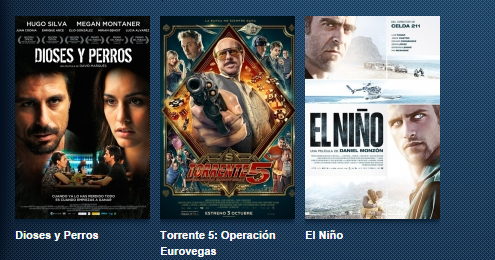
Guardar en una agenda toda la información de nuestros amigos, y crear un formulario que nos pida el nombre de uno y nos saca la información de él.

También existe en el formulario la posibilidad de visualizar toda la agenda.



24.

Realizar la siguiente web para un cine



De todas las películas que tenemos en nuestro cine, en la página inicial debemos presentar 3 que cambian aleatoriamente cada vez que nos conectamos al cine , de las 12 que tienen en el cine .Utilizar array \_rand para trabajar aleatoriamente

25.

Recuento de votos de votos de una elección.

Realizar el escrutinio y presentar los resultados de los países candidatos



La información de los votos la encontramos en votos.txt que tendrá una línea por candidato y por voto

Peru \n

España \n

Chile \n

Peru \n

Peru \n

Mexico \n

Para leer el fichero utilizamos la función file() que nos guarda el fichero en una matriz

Para visualizar las barras utilizamos verde.gif que contiene

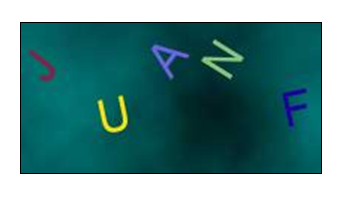


La información de los candidatos debe presentarse por orden alfabético y por orden de puntuación, utilizando para ello ksort() y arsort()

26.

Ejercicio:

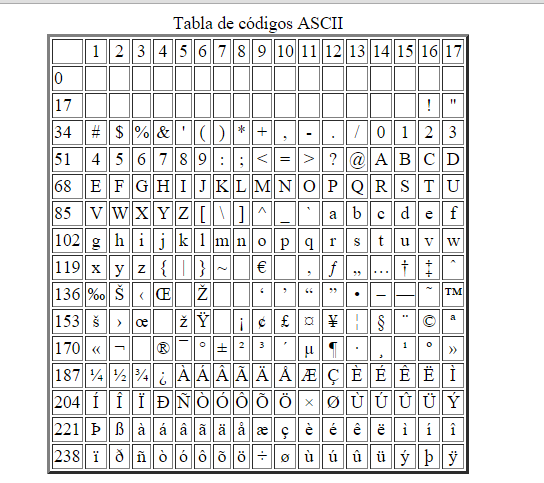
captcha?texto=uanfj



27. Utilizando funciones de cadenas hacer



28. Presentar



29. Validar un correo electrónico sin utilizar expresiones regulares

Comprobar que existe punto y @

Que solo haya una @

Que al menos exista un .

Se comprueba que sólo existan caracteres alfabéticos

30.

Cambiar el ejercicio 24 subiendo nosotros las películas de esta semana. Para lo cual presentamos un formulario que nos permite subir las imágenes

31.